Introduction to data analytics course project report

Amex Group Code "E"

Monday 2nd December, 2019

Introduction

In this project, we were provided with data(courtesy of American Express) which contains various predictors such as credit worthiness, annual income, etc(there were about 50 predictors), and a label indicating whether that person had defaulted or not. So, the challenge is given a new applicant, predict whether he/she defaults.

Preprocessing the data

We one-hot encoded the categorical variables(there was only one). We then converted all the "missing" and "na"(str type) values to np.nan, and coverted the entire dataframe to float values. The columns which had more than 50% missing values were removed, since they are unlikely to predict information.

Feature engineering

No imputation of missing values was done for boosting models since they can natively handle it and imputing using mean,median,MICE package, etc was only worsening the validation results. For random forest, logistic regression we used mean imputation. We then reduced the skewness of data and removed outliers using Sklearns's RobustScaler package. We added several new features which made financial sense(for instance "average-severity-default","due-income-ratio",etc are some of the added features). Feature importance as seen by LiteGBM model is shown in Fig 1, this figure is just qualitative and should not be taken seriously since we use an ensemble model at the end.

Individual Model building

The various individual classifiers we tried are:

1. XGBoost
2. LiteGBM
3. CatBoost
4. Random Forest

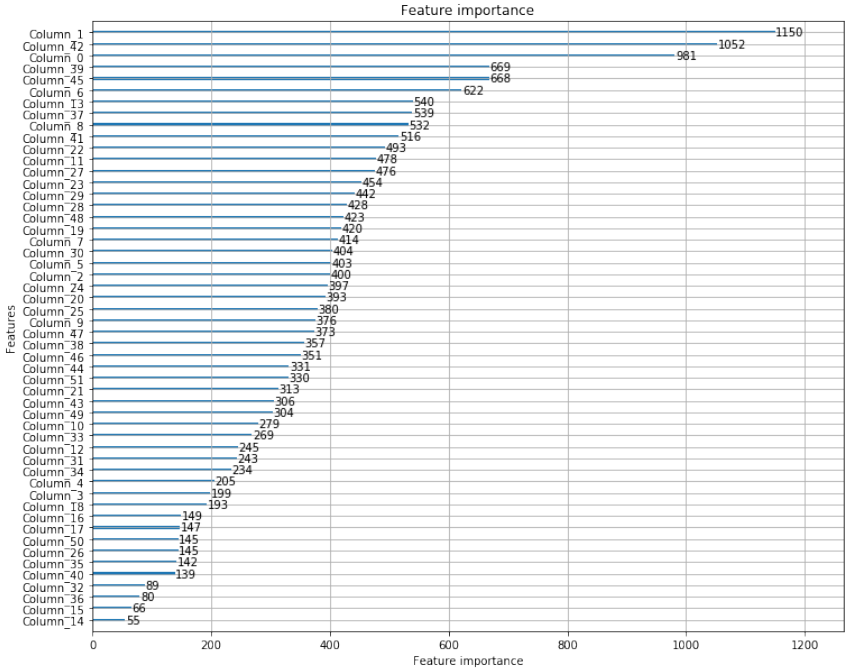

Figure 1: LightGBM feature importance-higher values indicate more importance

5. Logistic Regression
6. SVM

We individually tuned the hyperparameters of each model using Cross Validation. Each individual classifier by itself was able to achieve a good validation f1-score(about 0.57) except SVM which gave bad result. Inorder to achieve a good f1-score we also did thresholding of the probability such that if probability of class $1 > 0.25$, we predicted it as class 1 instead of the usual 0.5 threshold.

Ensemble Model

Ensemble learning is a machine learning technique whereby we can combine the various individual models we built to construct an even more powerful classifier. We used an ensemble of XGBoost, LiteGBM, CatBoost with averaged probability of Class 1 predicted by them with a threshold of 0.3 as the prediction metric(i.e. if average prob of class 1 predicted by the three classifiers is > 0.3 then predict as Class 1, else predict as Class 0). Adding random forest, logistic regression only lowered the validation score and the averaged probability method outperformed the majority-vote method.

Results

In the end we got(on our own test set):

1. f1-score = 0.593
2. gini-index = 0.58
3. balanced-accuracy = 0.72
4. auc-score = 0.79

Note: Above values might change slightly due to the random nature of splitting train and test data.
The metric we chose and optimised(via probability thresholding) is the f1-score because for a class imbalanced data set such as this problem blinding trusting accuracy would be misleading(suppose 90% of data is Class A and 10% is of class B, then a classifier which always predicts class A would have an accuracy of 90% which would then mislead us to believe that the above classifier is a good classifier). Especially for a bank false negatives(we predict that they don't default while they actually default) are extremely bad when compared to false positives(we predict that they default while they don't default) and this is reflected in the f1-score(decreases if we predict huge number of false negatives). Intuitively we observe that while the classifier has good accuracy, it has sub optimal f1-score which indicates false negatives. Lastly we were able to achieve a test f1-score of 0.6089 in the Amex leader-board and a leader-board rank of 1(as of today).